A routing strategy for non-cooperation wireless multi-hop ad hoc networks

Dung T. Tran*, Trang T.M. Truong and Thanh G. Le
Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam

Abstract. Choosing routes such that the network lifetime is maximized in a wireless network with limited energy resources is a major routing problem in wireless multi-hop ad hoc networks. In this paper, we study the problem where participants are rationally selfish and non-cooperative. By *selfish* we designate the users who are ready to tamper with their source-routing (*senders could choose intermediate nodes in the routing paths*) or next hop selection strategies in order to increase the total number of packets transmitted, but do not try to harm or drop packets of the other nodes. The problem therefore amounts to a non-cooperative game. In the works [2,6,19,23], the authors show that the game admits Nash equilibria [1]. Along this line, we first show that if the cost function is linear, this game has pure-strategy equilibrium flow even though participants have different demands. However, finding a Nash equilibrium for a normal game is computationally hard [9]. In this work, inspired by mixed-strategy equilibrium, we propose a simple local routing algorithm called MIxed Path Routing protocol (MiPR). Using analysis and simulations, we show that MiPR drives the system to an equilibrium state where selfish participants do not have incentive to deviate. Moreover, MiPR significantly improves the network lifetime as compared to original routing protocols.

Keywords: Local routing, Network lifetime, Nash equilibrium

1. Introduction

In recent years, the problem of lifetime maximization has been studied extensively. Since wireless devices have limited resources, e.g. battery, it is essential to develop energy efficient routing algorithms which optimize the overall energy utilization of the network. As done in [2,3], we define the lifetime of the network as the time before the first node on a forwarding path runs out of battery. From the literature, we know that the minimum energy path routing does not yield maximum lifetime of the network. The reason is that they may create congested areas which cause some nodes to quickly lose energy. Typically, the problem of routing for maximizing network lifetime is more important than the problem of finding a path with minimum energy consumption.

Many works have focused on finding energy-efficient routing for maximizing the network lifetime. However, they assume that all participants will follow the rules of the protocol. Since wireless ad hoc networks might have diverse participants, there may exist selfish participants who basically act for their own optimality. Commonly, the result of local optimization with conflicting interests does not lead to any type of global optimality. This is because the maximum amount of data a node can transmit depends not only on its chosen path but also the current traffic on that path. Senders without cooperation may choose the same set of preferred paths, leading to a paralysis of the system very soon.

---

*Corresponding author: Dung T. Tran, Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam. E-mail: ttdung@fit.hcmus.edu.vn.

In this work, we assume that all the participants' objective is to maximize the amount of their transmitted data before their routing paths are disconnected. Since participants select the routing paths to maximize the amount of transmitted data independently, the problem thus becomes one of a *noncooperative game* [1]. Because the total amount of data a participant can transmit depends on its chosen path and the number of participants choosing this path, this game hence is a *congestion game* [1]. Thus, we take the advantage of game theory into the design of the routing protocol.

Let us first recall some basic concepts of game theory. A normal game is defined by a tuple (*players, strategies, utility functions*), in which each player has a set of strategies to choose, called *strategy space*; a utility (*payoff*) function of a player takes as input a *strategy profile* (a specification of strategies for every player) and yields a representation of utility as its output. A *Nash equilibrium* of a game is the point where there exists a *strategy profile* which fully specifies all actions in the game such that no player could gain more by unilaterally changing its strategy. In a game there may exist many Nash equilibria, so that the *social profit* (sum of players' utilities) of the game at each Nash equilibrium may get different values.

The works [2,6,19,23] show that the general version of routing game admits Nash equilibria. In this work, we assume that nodes transmit data in packets. This means that the network flow is in integer as the model of Rosenthal [6]. The first part of this paper focuses on perspectives of the non-cooperative Lifetime Game. In this part, we summary prior results and present our results. We show that if cost function is linear, the *pure-strategy equilibrium flow* exists even through nodes have different demands.

Finding a Nash equilibrium in a normal game is complete for the complexity class PPAD [9], even in the case of two players [15], this is evidence of the intractability of the problem. Orda et al. [19] and Lewin et al. [2] show that if all participants choose the best response, i.e. the current maximum residual energy path[1], the systems will eventually converge to a Nash equilibrium. However, in the practical wireless ad hoc networks, in oder to find the maximum residual path frequently, a node needs to have both the global map of the network and the updated energy of the edges. This requirement, however, is practically impossible to fulfill. Therefore, in the second part of this work we design a new protocol which satisfies two conditions:

1. It can drive the network into a Nash equilibrium point.
2. It can apply to the practical noncooperative ad hoc networks.

2. Model and problem formulation

We assume that nodes can adjust their transmission radio to save the power when sending packets to its neighbors. The network is modeled by a directed graph $G = (V, E)$, where $n = |V|$ is number of nodes and $m = |E|$ is the number of edges. There are k source-destination pairs $(s_1, d_1), \ldots, (s_k, d_k)$. Each source node i has r_i data packets (integer variable) and wants to maximize its data transmitted. Here, we consider wireless ad hoc networks where:

– Data flow is *splittable* (be able to transmit through many paths at a time), and be transmitted in packets with integer size.
– Each source node may have different amount of data.

---

[1]The maximum remaining energy path.

Above assumptions make this model like the one in [6].

Let define an increasing function $c_e(x) : \mathbb{N}^+ \rightarrow \mathbb{R}^+$ as the cost function to transmit x packets through link e, i.e. every node which uses this link experiences this cost. The cost, here, presents the amount of energy consumed when a packet is transmitted through a link. Then, the cost of a path p is as:

$$c_p = \sum_{e \in p} c_e(x_e) x_e \tag{1}$$

where x_e is the number of packets passed through edge e.

Let us define *Lifetime Game* as a tuple (G, c, r), where $c = \{c_1, \ldots, c_m\}$, and $r = \{r_1, \ldots, r_n\}$. *Equilibrium flow* is now defined as follows.

Definition 1 *Given an instance (G, c, r), and a set of source-destination pairs, an equilibrium flow is an assignment of flow to routing paths in which no source node could decrease its cost by unilaterally changing its routing paths. Specifically, let x be the feasible flow for instance (G, c, r), if p is a path in an equilibrium flow then for any other path p'.*

$$c_p(x) \leqslant c_{p'}(x)$$

Before presenting our main results, we summary prior results of this problem.

3. Prior results

In 1952, Wardrop [8] has modeled unregulated traffic as network flow with all paths between a given source-destination pair having equal cost function. The equilibrium routing, thus, is called *Wardrop equilibrium*. Since then, many properties and extensions to this traffic model have been studied [11–21]. We may classify this problem based on characteristics of the flow. There are two main kinds: data is splittable and non-splittable.

3.1. Data is non-splittable

In this case, source nodes can only transmit data through a single path at a time instant. All works of this kind consider the data flow in continuous variable, and the results are also applicable to the case with data flow as discrete variable. Fotakis et al. [22] proved that the affine cost function is the necessary condition for existence of pure NE. Panagopoulou and Spirakis [23] showed that pure NE always exists for instances with uniform exponential cost function ($c_e(x) = \exp(x)$). Harks et al. [24] generalized this to non-uniform exponential cost functions of the form $c_e(f) = a_e e^{\phi f} + b_e$ for some $a_e, b_e, \phi \in \mathbb{R}$, where a_e and b_e may depend on link e, while ϕ must be equal for every link. Finally, Harks and Klimm [25] proved the sufficient condition for the existence of pure NE. In their work, authors show that cost function must be either an affine function or is of type $c_e(f) = a_e e^{\phi f} + b_e$. The price of anarchy in this case is at most 2.618 (see [9, p. 476]).

3.2. Data is splittable

In this case, source nodes can transmit data through many paths at a time instant. Most work of this kind consider the data flow in continuous variable. The problem always admits mix-strategy equilibrium flow by Nash theorem [10]. If the cost function is differentiable and convex then pure-strategy equilibrium

flow exists and is unique [9, p. 468]. Moreover, if the cost function is linear then the price of anarchy is at most 4/3 (see [9, p. 485]).

Rosenthal [6] considered this problem in integer flow. Author showed that pure-strategy equilibrium flow always exists if source nodes have *same* demand. If source nodes have different demand and link cost functions are general, the problem needs not admit pure-strategy equilibrium flow. In our work, we prove that if the cost function is linear, the problem has pure-strategy equilibrium flow even through source nodes have *different* demand. These equilibrium flow are not unique. The price of anarchy in this case is also at most 4/3.

The work of Levin-Eytan et al. [2] studies game-theoretic aspects of lifetime maximization problem. However, authors study the maximum lifetime problem in multicast and anycast routing. In this work, authors modeled a single multicast section as a static game, this thus is quite different from the model we discussed above.

4. Our results

4.1. Existence of pure-strategy nash equilibrium

Let a *strategy* of a player include:

1. A set of used paths from source to destination
2. Fraction of data will be transmitted on each path

In Lifetime Game, the set of paths between any pair of source-destination is finite, because there are finite number of nodes and finite number of links in the network. This means that Lifetime Game is a finite strategic game. Thus, it admits a mixed-strategy equilibrium flow (by Nash theorem [10]). However, the mix-strategy NE does *not* guarantee for an integer solution. Here, we show that this game also admits pure-strategy NE.

Theorem 1 *Given an instance of Lifetime Game where players have different demand and data flow is splittable. If the cost function is linearly increasing then Lifetime Game has pure-strategy equilibrium flow.*

Proof See detail in the appendix.

5. MiPR protocol

We now consider practical wireless ad hoc networks, where a single source node does not know either the global network topology, or other source nodes. Since the players now do not know the chosen paths of others, a node with the strategy of choosing the maximum residual energy path may even yield the worst case, e.g. the case in Fig. 1.

Assume every source node S_i, $i = \{1, \ldots, k\}$, wants to send data to destination T at the same time, and the capacities of edges (S_i, A_j) and $(B_i, T) >> (A_i, B_i)$ $\forall i, j$. For any source node S_i, the path (S_i, A_1, B_1, T) gives the lifetime value[2] k, all the other paths give the lifetime value $k/2$. Thus, all

---

[2]The total number of packets a path/link could transfer until one node in the path runs out of battery.
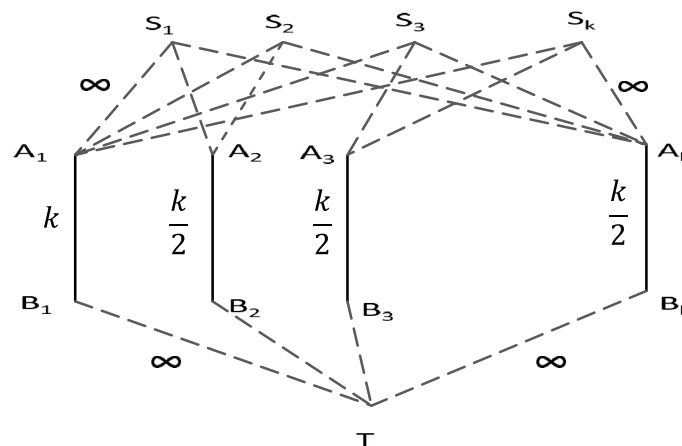
Fig. 1. Example of the worst case of non-cooperative game.

source nodes choose path (S_i, A_1, B_1, T) to send data, causing the network to transmit only 1 packet per node (because the capacity of edge (A_1, B_1) is k). The optimal solution for this case is to have each source S_i choose the path (S_i, A_i, B_i, T), so that each node could transmit at least $k/2$ packets. We note that, in this optimal solution, the utility of each node is not distributed fairly, since node S_1 can transmit up to k packets, while the other nodes can only transmit $k/2$ packets.

Typically, the central part of the network is likely to be the congested area because many transmission paths go through it. This not only causes nodes in the center to drain batteries faster, but also raises security issues. To overcome this problem, we make use of the concept of mixed strategies, i.e. source nodes will select each path in the set of available paths to destinations with some probability such that the network will converge to a mixed-strategy equilibrium flow. At this point, a single node cannot increase its lifetime by switching to another path. In other words, the network flow is balanced for all participants.

5.1. Problem of geographic routings

In the experiment of Mei and Stefa [3], the authors show that, in geographic routings,[3] the traffic load in the center area is always higher than that in the border area. This is the reason that nodes in the center area will be drained their energy soon. For example, nodes are uniformly distributed in a rectangle area. Source and destination nodes are both selected randomly. Then the traffic load in the network is as shown in Fig. 2.

5.2. Balanced energy routing protocol

In order to overcome the problem of geographic routings, we design a protocol, called *balanced energy routing protocol*. Nodes follow this protocol would not create any congested areas like in geographic routing.

The basic idea is that we transform the network area into a symmetric area which has no center part. For this purpose, we map the network area onto a sphere surface. First, we claim that the sphere

---

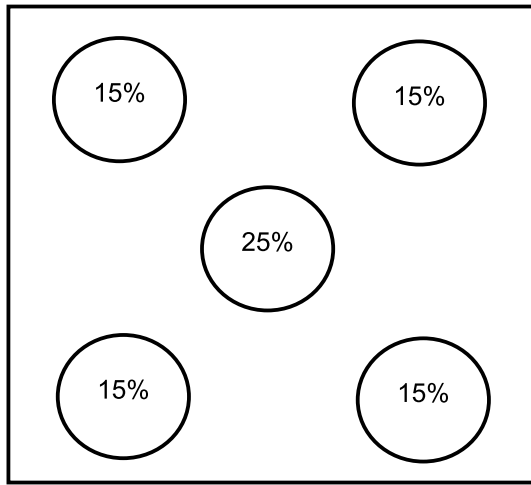[3]Select the closest node to the destination to be the next hop.

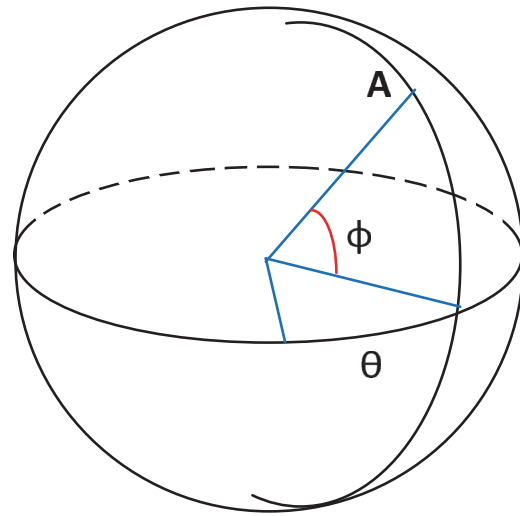Fig. 2. Point $A(\phi, \theta)$ in sphere surface.

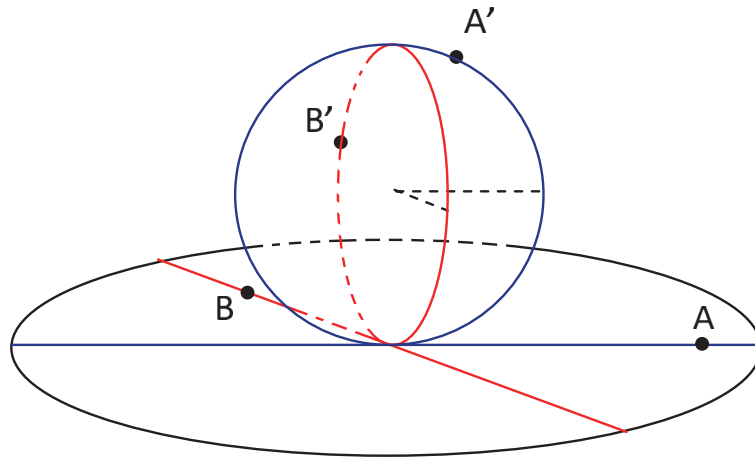Fig. 3. Sample traffic load in geographic routing.

Fig. 4. Points A', B' are images of A, B on sphere surface respectively.

surface satisfies the condition that there is no center part in the network. Because the sphere surface is a symmetric region, if nodes are uniformly distributed on it, we easily see that every node on the surface can be on the shortest direction of a source-destination pair with the same probability.

The transformation process just maps nodes into new locations on the sphere surface; however, it keeps all edges in the original graph. This means that neighbors of node u in the new region are exactly those of u in the original region. A point on the sphere surface can be defined by its latitude and longitude.

For example, point A in Fig. 3, A's latitude is the angle ϕ, and its longitude is the angle θ. To transform the mapping of a network from a circle area to a sphere surface, let the circle be a set of all lines of the same length with the center of the circle as their midpoint. We then bend those lines upwards to create a sphere. We can visualize the mapping process as shown in Fig. 4.

Another issue is that nodes on the sphere surface have to be random, and uniformly distributed. The circle area can be divided into n small layers with equal area. Then we also divide the sphere surface into n equal slices along vertical line, as in Fig. 5. The mapping process will map each layer of the circle
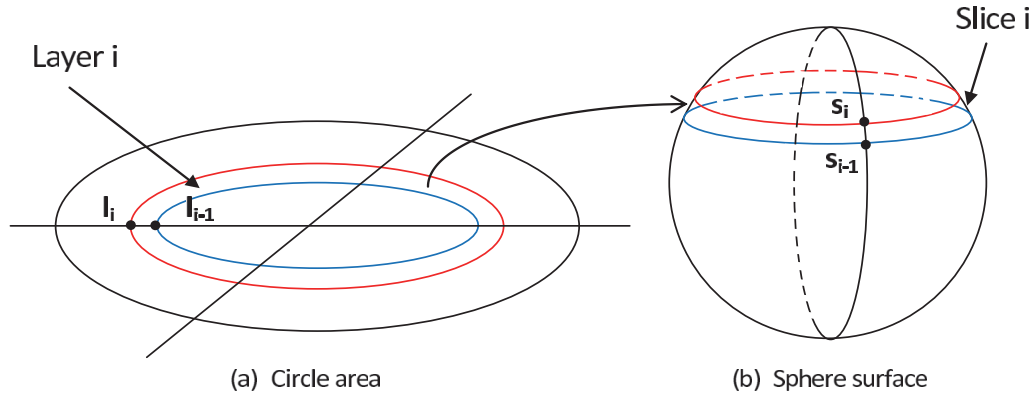
Fig. 5. Layer i^{th} of the circle region mapped to the corresponding slice i^{th} of the sphere surface.

region to its corresponding slice of the sphere surface. If node u belongs to the ith layer, then its image on the sphere belongs to the ith slice.

We now formally address the mapping process. Let a point $A(x, y)$ with coordinates (x, y) on the two-dimensional circle region with radius r. The center of the circle has coordinates (0,0). Let A' be the image of A on the sphere surface. We first find the formula to calculate the approximate longitude of A'. We can easily see that the longitude $\theta_{A'}$ of A' on the sphere surface is approximate the angle of A on the circle area, hence it can be calculated as follows:

$$\theta_{A'} = arctan\,(y, x) \tag{2}$$

The longitude is quite simple to find, but the latitude is more complicated. Assume the circle is divided into n equal layers which have the same center but different radius. Let layer i be the area between radii l_{i-1} and l_i, each layer be mapped to a corresponding slice i, which is the area between latitude s_{i-1} and s_i. Each area of layer i is $(\pi r^2)/n$. Given r and n, we can easily find the value of each l_i, taking $l_1 = 0$, as follows.

$$l_i = \sqrt{\frac{r^2}{n} + l_{i-1}^2} \tag{3}$$

We say a point $A(x, y) \in$ layer i^{th} if and only if $l_i \leqslant \sqrt{x^2 + y^2} < l_{i+1}$, i.e., the distance from A to the center is between l_i and l_{i+1}. We also divide the sphere into n slices, each slice have the same size in latitude π/n. Thus, slice i^{th} will be in range $[\frac{\pi}{n}(i - 1) - \frac{\pi}{2}, \frac{\pi}{n}i - \frac{\pi}{2}]$.

Assume that $A \in$ layer i^{th}, latitude $\phi_{A'}$ can be found by the following formula:

$$\phi_{A'} = \left[\frac{\pi}{n}(i - 1) - \frac{\pi}{2}\right] + \frac{\pi}{n}\frac{\sqrt{x^2 + y^2} - l_i}{l_{i+1} - li} \tag{4}$$

The distance between two points on the sphere surface with radius R is the shortest curve connecting them on the surface. We use Haversine's formula [4] for calculating the distance Δ between point $u(\phi_u, \theta_u)$ and $v(\phi_v, \theta_v)$ on the sphere surface:

$$\Delta = 2R.arcsin\sqrt{sin^2\left(\frac{\Delta_\phi}{2}\right) + cos\phi_u cos\phi_v sin^2(\frac{\Delta_\theta}{2})} \tag{5}$$
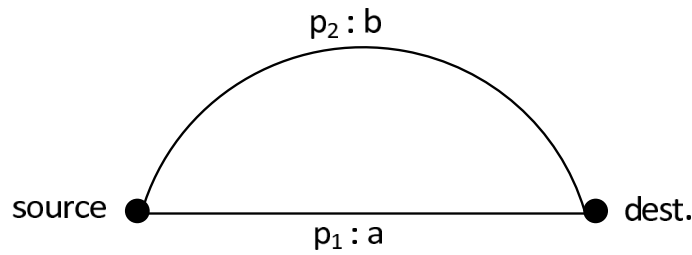
Fig. 6. Scenario of lifetime game.

On the sphere surface the logical distance between pairs of nodes changes, as all nodes near the border of the original area are now quite close together on the north area of the sphere. Therefore, a pair source-destination located on borders of opposite sides would prefer the routing path along the border rather than go through the central part. Thus, in the new domain, the shortest path between two nodes that are located near the border no longer goes through the congested center in the network. This path is clearly the shorter arc of the circle which contains the source and destination and is around the congested area.

We claim that if the nodes are uniformly distributed in the circle region, and if the size of each layer is small enough, then nodes are uniformly distributed on the sphere surface. We sketch the proof of our claim as follows. Since nodes are uniformly distributed, layers with the same size will contain an approximately equal number of nodes. We note that nodes are also uniformly distributed on each layer. Each layer is mapped to each slice on the sphere surface, hence, slices have approximately the same number of nodes. Slices and layers are similar in their shape (circular), and nodes in the same layer keep their relative distances. Nodes therefore are uniformly distributed in slices. Taking the limit, we obtain that nodes are uniformly distributed on the sphere surface.

We note that it is impossible to have a point-to-point mapping from a circle area to a sphere surface continuously. However, nodes are, in fact, distributed on the circle discretely. Therefore, we can have point-to-point mapping while approximately keeping the uniform distribution.

When the original region is a rectangle or any arbitrary shape in two-dimensional space, if we can scale them to an appropriate circle area in which nodes are randomly distributed, then we can apply this transformation to achieve a symmetric region.

5.3. MiPR protocol

The *balanced energy routing protocol* could not solve the problem of selfish nodes. If a node likes to reduce the delay by shortening its routing path, it might select the geographic routing without any penalty from the protocol. In this work, we assume that nodes don't have to strictly follow the protocol. Each single node may choose it routing paths for maximizing its utility. In this work, we propose MiPR protocol for this new issue.

We firstly model the network in the experiment above as a strategic game, called *Lifetime Game*. In this game, each source node is a player, and the set of paths from source to destination is a set of strategies. Consider two players, player 1 and player 2 are located at the same place on the border the network area, while their destinations are also located on the border, but at the other side of the network area. They share the same set of routing paths which could be divided into two groups: one group of paths created by geographic routing which will pass through the center network area, represented by path p_1 in Fig. 6;

Table 1
Lifetime game

		Player 2	
		p_1	p_2
Player 1	p_1	$a/2, a/2$	a, b
	p_2	b, a	$b/2, b/2$

the other group is created by *balanced energy routing protocol* which will go along the border of network area, represented by path p_2 in Fig. 6.

For a single transmission, path p_1, which goes through the center, has lifetime of a, and path p_2 has lifetime of b, $b < a$. Path p_2's lifetime is smaller because it is longer, with many nodes involved in the transmission. The total energy consumed then is larger. If two players choose path p_1 then p_1's lifetime is reduced to $a/2$. Or if they both choose path p_2, then p_2's lifetime is $b/2$. The matrix representing this game is shown in Table 1.

Lemma 1 If player 1 (player 2) chooses the path p_1 (p_2) with probability $q_1 = \frac{2a-b}{a+b}$ ($q_2 = \frac{2b-a}{a+b}$) then the other player could not improve its utility by changing its routing strategy.

The proof of Lemma 1 is shown in the following section.

5.3.1. Correctness of the protocol

Let q_1 and q_2 be the probability that player 1 and player 2 will choose path p_1 and p_2, respectively. The probability that player 1 and player 2 will transmit on path p_2 is $1 - q_1$, $1 - q_2$, respectively. Then the utility u_1 of player 1 is:

$$
\begin{aligned}
u_1 &= \frac{a}{2}q_1q_2 + aq_1(1 - q_2) + b(1 - q_1)q_2 + \frac{b}{2}(1 - q_1)(1 - q_2) \\
&= q_1 \left(\frac{2a - b}{2} - q_2 \frac{a + b}{2} \right) + \frac{b}{2}q_2 + \frac{b}{2}
\end{aligned}
\tag{6}
$$

Likewise, we have

$$
u_2 = q_2 \left(\frac{2a - b}{2} - q_1 \frac{a + b}{2} \right) + \frac{b}{2}q_1 + \frac{b}{2}
\tag{7}
$$

Consider Eq. (7), the utility u_2 does not depend on any probability q_2 if

$$
q_1 = \frac{2a - b}{a + b}
\tag{8}
$$

If $q_1 > \frac{2a-b}{a+b}$, the best response for player 2 is to set $q_2 = 0$. Otherwise, when $q_1 < \frac{2a-b}{a+b}$, the best response for player 2 is to set $q_2 = 1$. The best response of the two players can be graphically represented in Fig. 7. In the graphical representation, the best response function is the set of best response values when varying the other best response. Based on the concept of mutually best responses, we can identify the Nash equilibria as the crossing points of the best response functions in Fig. 7.

Therefore, if every node randomly chooses path p_1 and p_2 with probability $\frac{2a-b}{a+b}$, $\frac{2b-a}{a+b}$, respectively, then no single node can improve its lifetime by unilaterally deviating from this strategy.
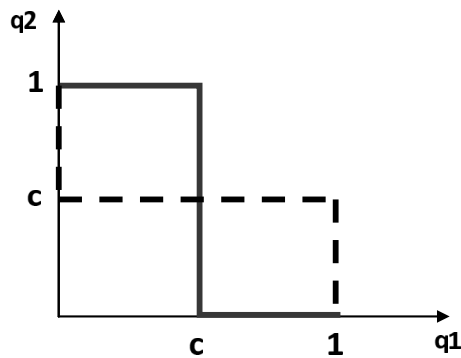
Fig. 7. The best response functions in Lifetime Game, q_1 as a function of q_2 and vice versa. The dash line represents the best response of player p_1, and the solid line is the best response of player p_2. Where $c = \frac{2a-b}{a+b}$.
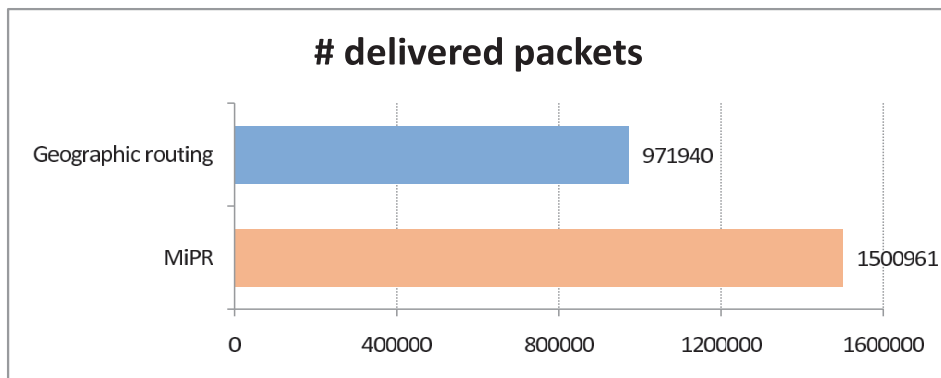

Fig. 8. Total number of packets delivered.

5.3.2. Protocol summary

In sum, MiPR follows two main steps:

1. Setting up path p_1, and path p_2.
2. Using mixed-strategy in Lifetime Game at Nash equilibrium to route data through these paths.

The next hop on path p_1 is the neighbor node which is closest to the destination in the two dimensional space. Similarly, the next hop on path p_2 is the neighbor node which is closest to the destination on the sphere surface. To find the probability of choosing a path, the sender node needs to know the value of a, b. Nodes could obtain the values of a, b by using request packets while setting up routing paths. The request packets will ask nodes about their energy capacity while finding routing paths. One could argue that nodes may act selfishly by lying in their response about their capacity, and this scheme therefore may fail. In this case, we could apply the truthful mechanism in [3] for finding the real value for the shortest paths p_1, p_2.

6. Simulation and evaluation

In order to assess the merit of our proposed protocol, we implement MiPR to compare it with other protocols in the same environment. We assume that the nodes know their position by being equipped
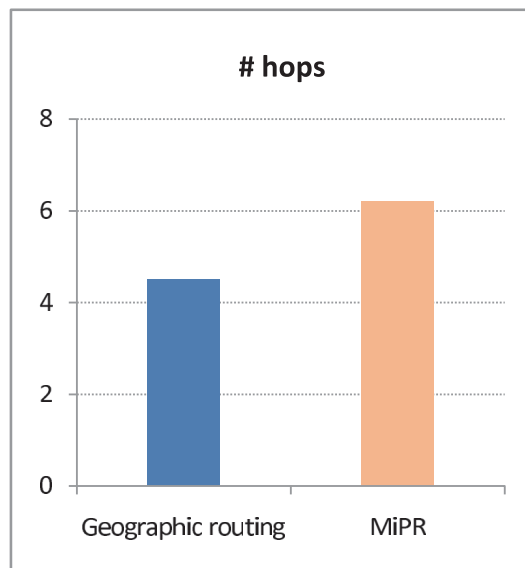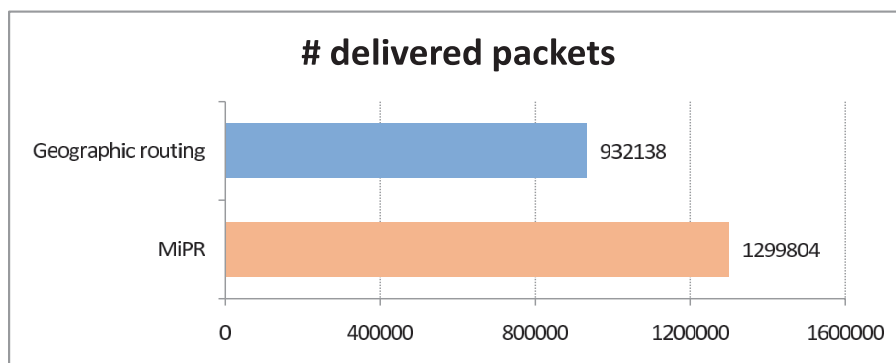
Fig. 9. Average hop-path length.

Fig. 10. Total number of packets delivered in the case of 300 nodes.

with a GPS (Global Positioning System) unit, or using some other location determination technique, such as the one in [5]. We consider the scenario of 500 nodes randomly and uniformly deployed on an area 1000×1000. Each node is equipped with a radio with the maximum transmission range of about 200. Nodes have the ability to adjust their transmission ranges to save energy. Thus, we assume nodes use the minimum energy required to transmit packets to the next hop. In *geographic routing*, we let the sender node choose the closest node to the destination as a next forwarder.

We initially assign each node equal battery energy. In each scenario, we ran over 1000 different topologies to get the average value. Figure 8 shows the simulation results of the protocols.

Simulation result shows that MiPR significantly improve network lifetime (1,500,951 vs 971,940 messages delivered), since it balances the traffic between paths, avoiding the creation of a congested area. There is, however, a different situation when we consider the average hop path length, see Fig. 9.

The average hop-length in MiPR is about 20% longer than that in Geographic Routing. This means that it takes more energy and a longer path to send a packet in MiPR protocol. On the other hand, for
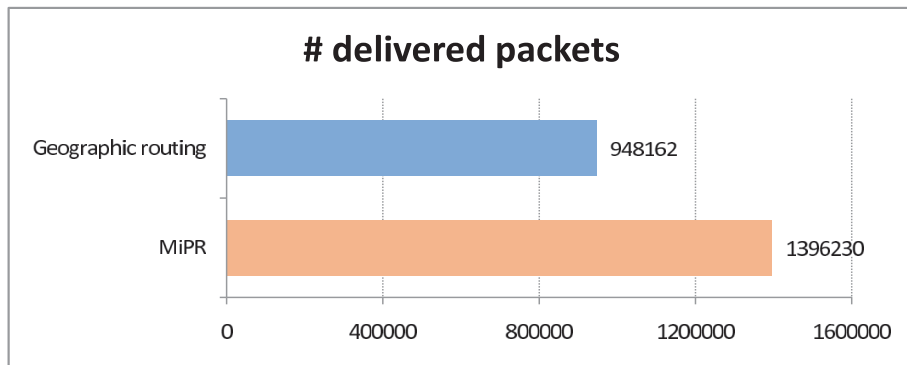
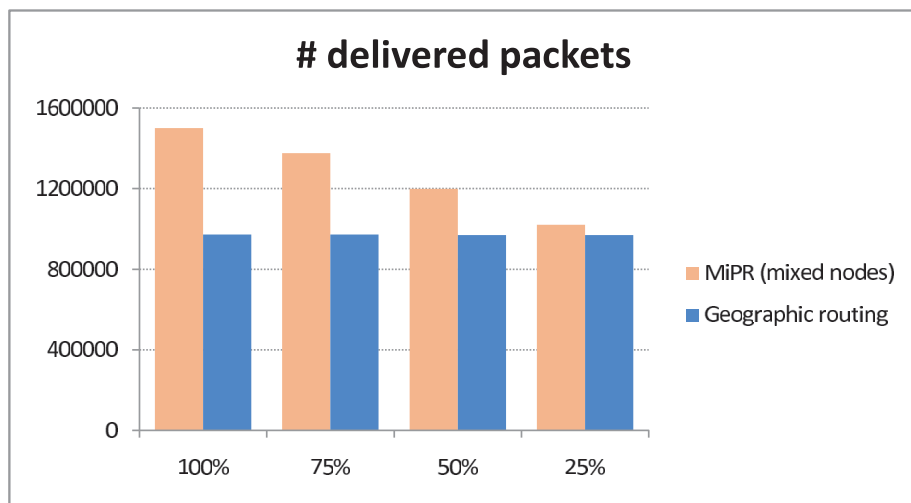Fig. 11. Total number of packets delivered in the case of 400 nodes.

Fig. 12. Total number of packets delivered when the network contains both selfish and regular nodes.

fairness and network lifetime MiPR performed better. This is a common trade-off in lifetime routing problem.

We now keep the same settings while changing the number of nodes. Figures 10 and 11 show the results of the case of 300 and 400 nodes, respectively. These results are quite similar to that of the case of 500 nodes. However, since the graph is not as dense as the case of 500 nodes, each node on the transmission path may have to spend more energy for forwarding the packets to the next hop. Thus, the network lifetime is reduced.

We consider the case in which the networks contain two groups of nodes, a group of selfish nodes (which can choose the routes) and the group that has to follow the geographic routing. Figure 12 shows the results of four different cases in which 100%, 75%, 50% and 25% of the total participants are selfish. We again see that the MiPR gives better results than the original routing protocol.

In this protocol, we assume the network starts operating with equal initial battery energy at each node. When nodes have different capacity of batteries, we just need to make a minor modification in the protocol. When it chooses the next hop in each path, it considers all neighbors that are closer to the destination, then chooses the node with maximum remaining energy.

7. Conclusion

We have studied the problem of routing to maximize the network lifetime in wireless multi-hop ad hoc networks where selfish nodes are taken into account. We first provide insight into the network lifetime problem through game-theoretic perspectives. We used strategic game model to deal with this problem. With affine cost function, we show that the game possesses pure-strategy equilibrium flow even through the senders have different demands.

In the second part, we have proposed a simple local routing algorithm for practical wireless ad hoc networks. Although MiPR does not give the optimal lifetime for the network, it drives the network to some fair equilibrium point where participants do not have incentive to deviate from the protocol. Moreover, MiPR significantly improves the network lifetime, in comparison with the previous routing protocols. This routing protocol is also applicable for other routing objectives of the same situation such as minimizing the network latency or numbers of packet lost.

Acknowledgments

Appendix

Proof of Theorem 1

We recall some notations used in this proof,

- n: number of vertices
- m: number of edges
- x_{ik}: number of packets node i transmit through edge k
- $x_k = \sum_{i=1}^{n} x_{ik}$: total packets pass through edge k
- $c_k(x) : \mathbb{N}^+ \to \mathbb{R}^+$: is an affine function of edge k, where x is a number of packets pass through it.
- r_i: data amount at i

We use *potential function method* to prove this theorem.

Claim 1 Any solution to the following problem is a pure-strategy Nash equilibrium:

$$Minimize \sum_{k=1}^{m} \left(\sum_{t=0}^{x_k} c_k(t)t \right) \tag{9}$$

$$subject\ to\ x_k = \sum_{i=1}^{n} x_{ik}$$

for $k = 1, \ldots, m$; and $x_{ik} \in [0, r_i]$ for $i = 1, \ldots, n$; $k = 1, \ldots, m$.

Proof. Since the solution to Eq. (9) always exist. It suffices to establish that each is an equilibrium. Let (x_{ik}^*) solve Eq. (9). Assume it does not give rise to an equilibrium. Then for some node j taking some strategy $\mathcal{A}$ under (x_{ik}^*) can reduce its cost by switching to some strategy $\mathcal{B}$.

Here, we define a strategy $\mathcal{A}$ for a node i as a description of:

- – Set of paths from i to its destination
- – Number of packets are transmitted on each path

We prove claim 1 from the simple case to the general case as follows:

- – *Case 1:* Node j moves one packet from path a to path b
- – *Case 2:* Node j moves $\delta(>1)$ packets from path a to path b
- – *Case 3:* Node j moves δ packets from path a to paths: $b_1, \ldots, b_l$ where each path b_i, $i = 1, \ldots, l$, is added δ_i packets which are taken from path a. We have $\delta = \sum_{i=1}^{l} \delta_i$.
- – *Case 4 (general):* Node j changes from strategy $\mathcal{A}$ to strategy $\mathcal{B}$.

Case 1: Node j moves *one* packet from path a to path b
Node j made this move, i.e.

$$\sum_{k \in b \setminus a} c_k(x_k^* + 1)(x_k^* + 1) < \sum_{k \in a \setminus b} c_k(x_k^*)x_k^* \tag{10}$$

Consider the new values

$$(x_{ik}^o) = \begin{cases} x_{ik}^* + 1 & \text{if } i = j, k \in b \setminus a \\ x_{ik}^* - 1 & \text{if } i = j, k \in a \setminus b \\ x_{ik^*} & \text{otherwise} \end{cases}$$

(x_{ik}^o) is clearly feasible for Eq. (9). The objective function evaluated at (x_{ik}^o) is:

$$\begin{aligned} Objective(1) &= \sum_{k \in b \setminus a} \sum_{t=0}^{x_k^* + 1} c_k(t)t + \sum_{k \in a \setminus b} \sum_{t=0}^{x_k^* - 1} c_k(t)t + \sum_{k \in (a \cap b) \cup (\overline{a} \cap \overline{b})} \sum_{t=0}^{x_k^*} c_k(t)t \\ &= \sum_{k=1}^{m} \sum_{t=0}^{x_k^*} c_k(t)t + \sum_{k \in b \setminus a} c_k(x_k^* + 1)(x_k^* + 1) - \sum_{k \in a \setminus b} c_k(x_k^*)x_k^* \\ &< \sum_{k=1}^{m} \sum_{t=0}^{x_k^*} c_k(t)t. \quad \text{A contradiction because of Eq. (10).} \end{aligned}$$

Case 2: Node j moves δ packets from path a to b.
Consider the new values after this move.

$$(x_{ik}^0) = \begin{cases} x_{ik}^* + \delta & \text{if } i = j, k \in b \setminus a \\ x_{ik}^* - \delta & \text{if } i = j, k \in a \setminus b \\ x_{ik}^* & \text{otherwise} \end{cases}$$

(x_{ik}^0) is clearly feasible for Eq. (9). The objective function evaluated at (x_{ik}^0) is:

$$Objective(2) = \sum_{k \in b \setminus a} \sum_{t=0}^{x_k^* + \delta} c_k(t)t + \sum_{k \in a \setminus b} \sum_{t=0}^{x_k^* - \delta} c_k(t)t + \sum_{k \in (a \cap b) \cup (\overline{a} \cap \overline{b})} \sum_{t=0}^{x_k^*} c_k(t)t$$

$$= \sum_{k=1}^{m} \sum_{t=0}^{x_k^*} c_k(t)t + \sum_{k \in b \setminus a} \sum_{t=x_k^*+1}^{x_k^*+\delta} c_k(t)t - \sum_{k \in a \setminus b} \sum_{t=x_k^*-\delta+1}^{x_k^*} c_k(t)t$$

For a contradiction, we wish to show that,

$$\sum_{k \in b \setminus a} \sum_{t=x_k^*+1}^{x_k^*+\delta} c_k(t)t < \sum_{k \in a \setminus b} \sum_{t=x_k^*-\delta+1}^{x_k^*} c_k(t)t \qquad (11)$$

We note that since cost function is an increasing function, if a node cannot decrease its cost by moving one packet from path a to path b. i.e.

$$\sum_{k \in b \setminus a} c_k(x_k^* + 1)(x_k^* + 1) \geqslant \sum_{k \in a \setminus b} c_k(x_k^*)x_k^* \qquad (12)$$

then it cannot reduce cost by moving some δ (> 1) packets from path a to path b, because

$$\sum_{k \in b \setminus a} c_k(x_k^* + \delta)(x_k^* + \delta) \geqslant \sum_{k \in b \setminus a} c_k(x_k^* + 1)(x_k^* + 1) \geqslant \sum_{k \in a \setminus b} c_k(x_k^*)x_k^* \qquad (13)$$

Otherwise, there are two cases: (i). every packet that node j moves from path a to b induces a lower cost for j; (ii). node j can decrease its cost by moving up to δ packets from path a to path b, but not β packets, for some $\beta > \delta$.

In the subcase (i), this means,

$$\sum_{k \in b \setminus a} c_k(x_k^* + 1)(x_k^* + 1) < \sum_{k \in a \setminus b} c_k(x_k^*)x_k^* \qquad (14)$$

$$\vdots$$

$$\sum_{k \in b \setminus a} c_k(x_k^* + \delta)(x_k^* + \delta) < \sum_{k \in a \setminus b} c_k(x_k^* - \delta + 1)(x_k^* - \delta + 1) \qquad (15)$$

$$\Rightarrow \sum_{k \in b \setminus a} \sum_{t=x_k^*+1}^{x_k^*+\delta} c_k(t)t < \sum_{k \in a \setminus b} \sum_{t=x_k^*-\delta+1}^{x_k^*} c_k(t)t \qquad (16)$$

This implies Eq. (11) hold ((16) $\equiv$ (11)).

In the subcase (ii), because node j deceases its cost by moving δ packets from path a to path b, but not β packets, for some $\beta > \delta$, we have,

$$\sum_{k \in b \setminus a} (c_k(x_k^* + \delta)(x_k^* + \delta) - c_k(x_k^*)x_k^*) < \sum_{k \in a \setminus b} (c_k(x_k^*)x_k^* - c_k(x_k^* - \delta)(x_k^* - \delta)) \qquad (17)$$

and,

$$\sum_{k \in b \setminus a} (c_k(x_k^* + \beta)(x_k^* + \beta) - c_k(x_k^*)x_k^*) \geqslant \sum_{k \in a \setminus b} (c_k(x_k^*)x_k^* - c_k(x_k^* - \beta)(x_k^* - \beta)) \qquad (18)$$

Since $c_k(x)x$ is a linear function, $\sum c_k(x)x$ is also linear. Thus, Eqs (17) and (18) imply that $\sum_{k \in b \setminus a} c_k(x_k^*)x_k^*$ is linear and greater than $\sum_{k \in a \setminus b} c_k(x_k^*)x_k^*$ for some $x \geqslant x_k^* + \beta$. And $\sum_{k \in b \setminus a} c_k(x_k^*)x_k^*$ is linear and smaller than $\sum_{k \in a \setminus b} c_k(x_k^*)x_k^*$ for some $x \leqslant x_k^* + \delta$. (***)

Moreover, since node j transfers δ packets from path a to b, this means,

$$\sum_{k \in b \setminus a} c_k(x_k^* + \delta)(x_k^* + \delta) < \sum_{k \in a \setminus b} c_k(x_k^*)x_k^* \tag{19}$$

together with (***) implies,

$$\sum_{k \in b \setminus a} c_k(x_k^* + \delta - i)(x_k^* + \delta - i) < \sum_{k \in a \setminus b} c_k(x_k^* - i)(x_k^* - i), \forall i \in (0, \delta)$$

Then Eq. (11) holds. Therefore,

$$Objective(2) < \sum_{k=1}^{m} \sum_{t=0}^{x_k^*} c_k(t)t.$$

Case 3: Node j moves δ packets from path a to paths: $b_1, \ldots, b_l$ where each path b_i, $i = 1, \ldots, l$, is added δ_i packets which are taken from path a. We have $\delta = \sum_{i=1}^{l} \delta_i$.

If j can reduce its cost by moving δ_1 packets to path b_1 then it contradicts to proof of case 2. Otherwise, j does not make this move and it could reduce more cost. Likewise, for any other move δ_i to path b_i, node j cannot reduce its cost either. Therefore, this case is not true.

Case 4 (general): Node j changes from strategy $\mathcal{A}$ to strategy $\mathcal{B}$. In other words, j moves some packets from some of its current paths to some other paths.

First, consider path x_1 in j's current paths, assume that node j moves δ packets from path x_1 to paths $\{y_1, \ldots, y_k\}$ each path y_i with δ_i packets. Then this is case 3, i.e. j cannot reduce its cost. Likewise, j cannot reduce its cost on any of its other paths. In other words, node j cannot reduce its cost by unilaterally changing its strategy. □

Claim 1 immediately implies Theorem 1. □

References

[1] M.J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA: The MIT Press, 1994.
[2] L. Lewin-Eytan, J. Naor and A. Orda, Maximum-lifetime routing: system optimization & game-theoretic perspectives. MobiHoc '07. ACM, New York.
[3] A. Mei and J. Stefa, Routing in outer space: fair traffic load in multi-hop wireless networks. MobiHoc '08. ACM, New York, 2008.
[4] R.W. Sinnott, Virtues of the Haversine, *Sky and Telescope* **68**(2) (1984).
[5] J. Hightower and G. Borriello, Location systems for ubiquitous computing, *IEEE Computer* **34** (August 2001), 57–66.
[6] R.W. Rosenthal, The network Equilibrium Problem in Integers. By John Wiley & Sons, Inc. 1973.
[7] R.W. Rosenthal, A class of games possessing pure-strategy Nash equilibria, *International Journal of Game Theory* (1973), 65–67.
[8] J.G. Wardrop, Some theoretical aspects of road traffic research. In proceedings of the Institute of Civil Engineers. Pt. II Vol1, 1952, pp. 325–378.
[9] N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, 2007.
[10] J. Nash, Noncoperative games, *Annals of Mathematics* **54** (1951), 286–295.

[11] H.Z. Aashtiani and T.L. Magnanti, Equilibria on a congested transportation network. SIAM Journal on Algebraic and Discrete Methods, 1981, pp. 213–226.

[12] S. Dafermos, Traffic equilibrium and variational inequalities, Transportation Science, 1980, pp. 42–54.

[13] M. Florian, Nonlinear cost network models in transportation analysis, *Mathematical Programming Study* (1986), 167–196.

[14] M.A. Hall, Properties of the equilibrium state in transportation networks, Transportation Science, 1978, pp. 208–216.

[15] X. Chen and X. Deng, Settling the complexity of 2-player Nash-equilibrium, Electronic Colloquium on Computational Complexity, Fdns. Comp., USA, 2006.

[16] A. Nagurney, Sustainable Transportation Networks. Edward Elgar, 2000.

[17] Y. Nesterov, Stable flows in transportation networks. CORE Discussion Paper 9907, 1999.

[18] Y. Nesterov and A. De Palma, Stable dynamics in transportation systems. CORE Discussion Paper 00/27, 2000.

[19] A. Orda, R. Rom and N. Shimkin, Competitive routing in multi-user communication networks, IEEE/ACM Transactions on Networking, 1993, pp. 510–521.

[20] Y. Sheffi, Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall, 1985.

[21] M.J. Smith, The existence, uniqueness and stability of traffic equilibria. Transportation Research, 1978, pp. 419–422.

[22] S.K. Dimitris Fotakis and P. Spirakis, *Selfish unsplittable flows. Theoretical Computer Science*, 2005, pp. 226–239.

[23] P.N. Panagopoulou and P.G. Spirakis, Algorithms for pure Nash equilibria in weighted congestion games, *Journal on Experimental Algorithmics* (2006).

[24] M.K. Tobias Harks and R.H. Möhring, Characterizing the existence of potential functions in weighted congestion games. In Proc. of the 2nd Symposium on Algorithmic Game Theory (SAGT), 2009, pp. 97–108.

[25] T. Harks and M. Klimm, Characterizing the existence of pure Nash equilibrium in weighted congestion games. Accepted to the 37th International Colloquium on Automata, Languages and Programming (ICALP), 2010.

Dung T. Tran received his Master degree in Computer Science at the SUNY at Buffalo in 2006 and received his PhD in Computer Science at the University of Texas at Dallas in 2010. His research interests focus on the line between computer networks and game theory.

Trang T.M. Truong received her Bachelor and Master degree in Computer Science at the University of Science in 2004 and 2008, respectively. Her research interests focus on the network routing protocols.

Thanh G. Le received his bachelor degree in Computer Science at the University of Science in 2009. He is working toward his Master degree in Computer Science.